

# R-TVM, a polyhedral mapper for TVM

Klint Qinami, James Gilles, Muthu Baskaran, Sandeep Polisetty and Benoît Meister

qinami@reservoir.com

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0019522.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

TVMcon - 12/16/2021

# Relay/Halide vs Polyhedral: a fraternal battle

Started out as an image processing battle

- Halide[H13] vs PolyMage[P15]

Transferred to ML optimization

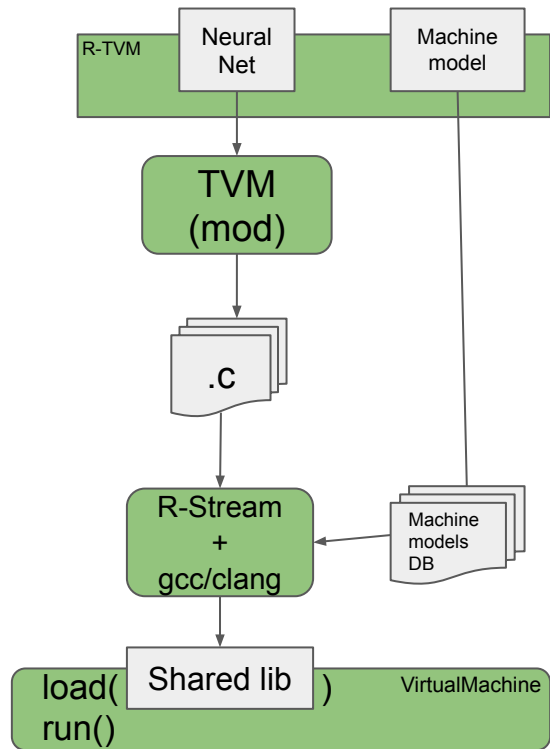
- TVM [T18] vs R-Stream.TF [R17], followed by several efforts including by Amazon, Facebook, Google, Huawei, Intel, NVIDIA, and academia.

Polyhedral:

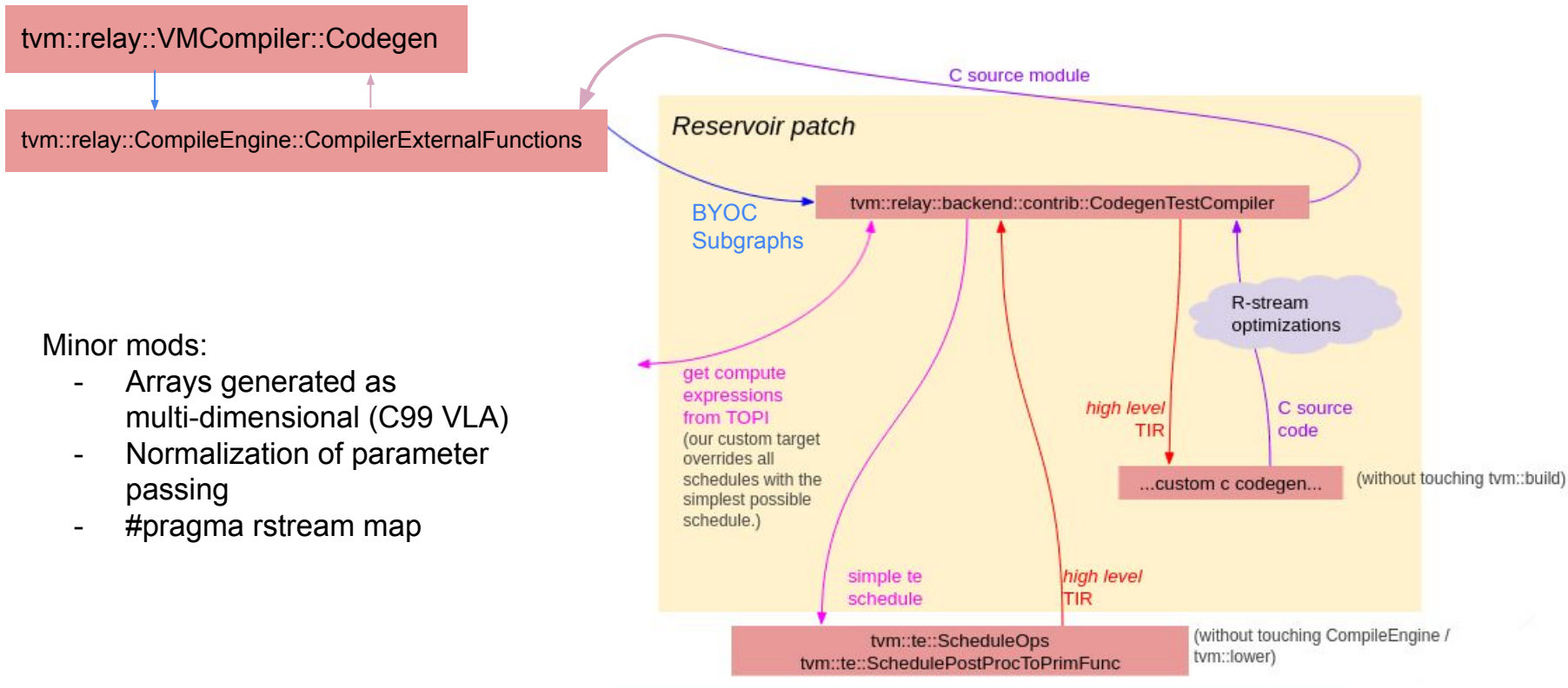
- Relies more on (potentially costly) modeling, less on autotuning
- Controversial claims about expressiveness, transformations portfolio, ...

# Relay front-end, Polyhedral backend

- Closest approach is Huawei's AKG[A21]
- Let TVM take care of the subgraph partitioning problem
  - TVM defines which layers should be mapped conjointly
  - Can use Relay to do some transformations (mostly unused)
- Let polyhedral map each subgraph
  - Subgraph codegen'd to C (next slide)
  - Polyhedral optimization to target



# Codegen: mod to the BYOC path



## Minor mods:

- Arrays generated as multi-dimensional (C99 VLA)
- Normalization of parameter passing
- `#pragma rstream map`

# Preliminary results & future work

Correct x86 results on a BERT proxy and ResNet18 using OpenMP

- Demonstrates proof-of-concept

Future experiments: tuning, more mixing, more targets & networks

- Let Relay optimize some layers, R-Stream others
- CUDA, OpenCL
- More NLP, Vision
- Recommendation models

# References

- [A21] Jie Zhao et al., “AKG: Automatic Kernel Generation for Neural Processing Units using Polyhedral Transformations,” PLDI '21, Virtual, Canada, June 2021.
- [H13] Jonathan Ragan-Kelley et al. “Halide: a language and compiler for optimizing parallelism, locality, and recomputation in image processing pipelines.” SIGPLAN Not. 48, no. 6 (June 23, 2013): 519.
- [P15] Ravi Teja Mullapudi et al. “PolyMage: Automatic Optimization for Image Processing Pipelines.” 20th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2015), Mar 2015.
- [R17] Benoit Pradelle et al. Polyhedral Optimization of Tensorflow Computation Graphs, International Workshop on Extreme-Scale Programming Tools (ESPT), 2017.
- [T18] Tianqi Chen et al. “TVM: An Automated End-to-End Optimizing Compiler for Deep Learning.” In Proc. of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2018